B 7

```
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFFFFFFFFFFFFFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNNNNN      NNN        III        CCC              NNNNNN      NNN  FFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN   NNN   NNN        III        CCC              NNN   NNN   NNN  FFFFFFFFFFFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN      NNNNNN        III        CCC              NNN      NNNNNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN        III        CCC              NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
NNN         NNN     IIIIIIIII        CCCCCCCCCCCC  NNN         NNN  FFF
```

**F

```
                                                    D  4
**FILE**ID**CNFSTORE




  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  TTTTTTTTTT    OOOOOO    RRRRRRRR    EEEEEEEEEE
  CCCCCCCC  NN      NN  FFFFFFFFFF    SSSSSSSS  TTTTTTTTTT    OOOOOO    RRRRRRRR    EEEEEEEEEE
CC          NN      NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NN      NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NNNN    NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NNNN    NN  FF          SS              TT      OO      OO  RR      RR  EE
CC          NN  NN  NN  FFFFFFFF      SSSSSS        TT      OO      OO  RRRRRRRR    EEEEEEEE
CC          NN  NN  NN  FFFFFFFF      SSSSSS        TT      OO      OO  RRRRRRRR    EEEEEEEE
CC          NN    NNNN  FF                  SS      TT      OO      OO  RR  RR      EE
CC          NN    NNNN  FF                  SS      TT      OO      OO  RR  RR      EE
CC          NN      NN  FF                  SS      TT      OO      OO  RR    RR    EE              ....
CC          NN      NN  FF                  SS      TT      OO      OO  RR    RR    EE              ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS        TT        OOOOOO    RR      RR  EEEEEEEEEE      ....
  CCCCCCCC  NN      NN  FF          SSSSSSSS        TT        OOOOOO    RR      RR  EEEEEEEEEE      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %TITLE  'DECnet Ethernet Configurator Module'
0 MODULE CNFSTORE        (
0                        LANGUAGE (BLISS32),
0                        IDENT = 'V04-000'
0                        ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     DECnet Configurator Module (NICONFIG)
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the routines for reading and storing the
1 !       system ID messages that are periodically broadcast on the NI.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  13-Oct-1982
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-002 TRC0002         Terry Cassidy           Aug-23-1984
1 !               Inhibit processing of pad bytes in Sys ID
1 !
1 !       V03-001 RPG0001         Bob Grosso              May-19-1983
1 !               Correct the arguements to LIB$SIGNAL.
1 !
1 !--
```

```
0054   1 %SBTTL  'Definitions'
0055   1
0056   1 !
0057   1 ! INCLUDE FILES:
0058   1 !
0059   1
0060   1 LIBRARY 'SYS$LIBRARY:STARLET';                    ! VMS common definitions
0061   1
0062   1 LIBRARY 'SHRLIB$:NMALIBRY';                       ! NICE code definitions
0063   1
0064   1 REQUIRE 'LIB$:CNFDEF.R32';
0155   1
0156   1 REQUIRE 'SRC$:CNFPREFIX.REQ';
0253   1
0254   1
0255   1 !
0256   1 ! BUILTIN functions
0257   1 !
0258   1
0259   1 BUILTIN
0260   1     INSQUE,                       ! INSQUE instruction
0261   1     REMQUE;                       ! REMQUE instruction
0262   1
0263   1 GLOBAL LITERAL
0264   1     SYSIDM_BUFSIZ = 64,
0265   1     ADRTYP_BUFSIZ = 14;
0266   1
0267   1 !
0268   1 ! TABLE OF CONTENTS:
0269   1 !
0270   1
0271   1 FORWARD ROUTINE
0272   1
0273   1         CNF$READ_SYSIDM,          ! Read the system ID messages from the NI
0274   1         BUFFER_ID : NOVALUE,      ! Buffer the ID messages and re-issue read
0275   1         STORE_ID : NOVALUE,       ! Partially parse the messages and store them
0276   1         INSERT_SID;               ! Place the ID messages in the circuit block
0277   1
0278   1 !
0279   1 ! EXTERNAL REFERENCES:
0280   1 !
0281   1
0282   1 EXTERNAL ROUTINE
0283   1
0284   1 !   Module CNFMAIN
0285   1
0286   1         CNF$EXIT,                 ! Clean up and exit
0287   1         CNF$TRACE,                ! Log messages to log file
0288   1         CNF$GET_ZVM,              ! Get zeroed virtual memory
0289   1         CNF$FREE_VM,              ! Free virtual memory
0290   1
0291   1 !   Module CNFWORKQ
0292   1
0293   1         WKQ$ADD_WORK_ITEM;        ! Add work to work queue
0294   1
0295   1 EXTERNAL ROUTINE
0296   1
```

```
:  112        0297  1          STR$COMPARE     : ADDRESSING_MODE (GENERAL);
:  113        0298  1
:  114        0299  1
:  115        0300  1 EXTERNAL LITERAL
:  116        0301  1
:  117        0302  1          CNF$C_ASYNCH_EFN;
:  118        0303  1
:  119        0304  1
:  120        0305  1 EXTERNAL LITERAL
:  121        0306  1
:  122        0307  1          CNF$_LOGIC,       ! Program logic error or unexpected condition
:  123        0308  1          CNF$_SYSID;       ! Error while obtaining system ID message
:  124        0309  1
:  125        0310  1 EXTERNAL
:  126        0311  1
:  127        0312  1          CNF$GQ_CIRSURLST : VECTOR [2];  ! List of circuits under surveillance
:  128        0313  1
```

```
  1 %SBTTL  'CNF$READ_SYSIDM  Read the system id messages on the NI'
  1 GLOBAL ROUTINE CNF$READ_SYSIDM (CIR) =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !   CNF$READ_SYSIDM issues the QIO to read System ID Messages on the NI.
  1 !
  1 ! FORMAL PARAMETERS:
  1 !
  1 !       cir     CIRcuit control block.  Contains buffers and information
  1 !               on the circuit.
  1 !
  1 ! IMPLICIT INPUTS:
  1 !       NONE
  1 !
  1 ! IMPLICIT OUTPUTS:
  1 !       NONE
  1 !
  1 ! ROUTINE VALUE:
  * ! COMPLETION CODES:
          Always return success

  1 ! SIDE EFFECTS:
  1 !       NONE
  1 !
  1 !--
  1
  2     BEGIN
  2     MAP
  2         CIR : REF BBLOCK;
  2     LOCAL
  2         STATUS;
  2
  2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
  2         $DESCRIPTOR ('cnf$read_sysidm'));
  2
  2     !
  2     !   Allocate the two buffers to hold the system ID message
  2     !   and the message header.  If the buffers are already there
  2     !   then a logic error of some sort has occurred.
  2     !
  2     IF  (.CIR [CIR$L_SYSIDMBUF] NEQ 0) OR
  3         (.CIR [CIR$L_ADRTYPBUF] NEQ 0)
  2     THEN
  2         SIGNAL (CNF$_LOGIC);
  2     EXECUTE (CNF$GET_ZVM (%REF(SYSIDM_BUFSIZ), CIR [CIR$L_SYSIDMBUF]) );
  2     EXECUTE (CNF$GET_ZVM (%REF(ADRTYP_BUFSIZ), CIR [CIR$L_ADRTYPBUF]) );
  2
  2
  2     CH$FILL (0, 8, CIR [CIR$W_IOSB]);   ! Initialize the I/O status block
  2
  2 !
  2 !   Issue QIO to read system id messages being broadcast on the NI
  2 !
P 2     STATUS = $QIO
P 2         (
;  187        P 0371  2             FUNC = IO$_READLBLK,
;  188        P 0372  2             CHAN = .CIR [CIR$W_CHAN],
;  189        P 0373  2             EFN = CNF$C_ASYNCH_EFN,
;  190        P 0374  2             IOSB = CIR [CIR$W_IOSB],
;  191        P 0375  2             ASTADR = BUFFER_ID,
;  192        P 0376  2             ASTPRM = .CIR,
;  193        P 0377  2             P1 = .CIR [CIR$L_SYSIDMBUF],
;  194        P 0378  2             P2 = SYSIDM_BUFSIZ,
;  195        P 0379  2             P5 = .CIR [CIR$L_ADRTYPBUF]
;  196          0380  2             );
;  197          0381  2
;  198          0382  2         IF NOT .STATUS THEN SIGNAL (CNF$_SYSID, 0, .STATUS);
;  199          0383  2         RETURN TRUE;
;  200          0384  1         END;                                    ! routine CNF$READ_SYSIDM


                                                         .TITLE  CNFSTORE DECnet Ethernet Configurator Module
                                                         .IDENT  \V04-000\

                                                         .PSECT  $PLIT$,NOWRT,NOEXE,2

                              45  43  41  52  54  00000 P.AAB:  .ASCII  \TRACE\                                    ;
                                                  00005         .BLKB   3
                                       00000005   00008 P.AAA:  .LONG   5                                          ;
                                       00000000'  0000C         .ADDRESS P.AAB                                     ;
6D  64  69  73  79  73  5F  64  61  65  72  24  66  6E  63  00010 P.AAD:  .ASCII  \cnf$read_sysidm\                ;
                                                  0001F         .BLKB   1
                                       0000000F   00020 P.AAC:  .LONG   15                                         ;
                                       00000000'  00024         .ADDRESS P.AAD                                     ;

                                                 SYSIDM_BUFSIZ==     64
                                                 ADRTYP_BUFSIZ==     14
                                                         .EXTRN  CNF$EXIT, CNF$TRACE
                                                         .EXTRN  CNF$GET_ZVM, CNF$FREE_VM
                                                         .EXTRN  WKQ$ADD_WORK_ITEM
                                                         .EXTRN  STR$COMPARE, CNF$C_ASYNCH_EFN
                                                         .EXTRN  CNF$_LOGIC, CNF$_SYSID
                                                         .EXTRN  CNF$GQ_CIRSURLST
                                                         .EXTRN  SYS$QIO

                                                         .PSECT  $CODE$,NOWRT,2

                                       00FC  00000         .ENTRY  CNF$READ_SYSIDM, Save R2,R3,R4,R5,R6,R7    ; 0315
                    57  00000000G  00   9E   00002         MOVAB   LIB$SIGNAL, R7                              ;
                    5E             04   C2   00009         SUBL2   #4, SP                                      ;
                           0000'   CF   9F   0000C         PUSHAB  P.AAC                                       ; 0349
                           0000'   CF   9F   00010         PUSHAB  P.AAA                                       ; 0348
                                   01   DD   00014         PUSHL   #1                                          ;
               0000G  CF           03   FB   00016         CALLS   #3, CNF$TRACE                               ;
                    56      04     AC   D0   0001B         MOVL    CIR, R6                                     ; 0356
                            38     A6   D5   0001F         TSTL    56(R6)                                      ;
                                   05   12   00022         BNEQ    1$                                          ;
                            3C     A6   D5   00024         TSTL    60(R6)                                      ; 0357
                                   09   13   00027         BEQL    2$                                          ;
                       00000000G   8F   DD   00029 1$:     PUSHL   #CNF$_LOGIC                                 ; 0359
                    67             01   FB   0002F         CALLS   #1, LIB$SIGNAL                              ;
                            38     A6   9F   00032 2$:     PUSHAB  56(R6)                                      ; 0360
```

```
                        04   AE       40   8F  9A 00035          MOVZBL  #64, 4(SP)                       ;
                                      04   AE  9F 0003A          PUSHAB  4(SP)                            ;
                     0000G   CF            02  FB 0003D          CALLS   #2, CNF$GET_ZVM                  ;
                             56            50  E9 00042          BLBC    STATUS, 4$                       ;
                                      3C   A6  9F 00045          PUSHAB  60(R6)                           ; 0361
                        04   AE            0E  D0 00048          MOVL    #14, 4(SP)                       ;
                                      04   AE  9F 0004C          PUSHAB  4(SP)                            ;
                     0000G   CF            02  FB 0004F          CALLS   #2, CNF$GET_ZVM                  ;
                             44            50  E9 00054          BLBC    STATUS, 4$                       ;
   08            00          6E            00  2C 00057          MOVC5   #0, (SP), #0, #8, 12(R6)         ; 0364
                                      0C   A6     0005C                                                   ;
                                           7E  D4 0005E          CLRL    -(SP)                            ; 0380
                                      3C   A6  DD 00060          PUSHL   60(R6)                           ;
                                           7E  7C 00063          CLRQ    -(SP)                            ;
                             7E       40   8F  9A 00065          MOVZBL  #64, -(SP)                       ;
                                      38   A6  DD 00069          PUSHL   56(R6)                           ;
                                           56  DD 0006C          PUSHL   R6                               ;
                                   0000V   CF  9F 0006E          PUSHAB  BUFFER_ID                        ;
                                      0C   A6  9F 00072          PUSHAB  12(R6)                           ;
                                           21  DD 00075          PUSHL   #33                              ;
                             7E       14   A6  32 00077          CVTWL   20(R6), -(SP)                    ;
                                000000000G 8F  DD 0007B          PUSHL   #CNF$C_ASYNCH_EFN                ;
                  00000000G  00            0C  FB 00081          CALLS   #12, SYS$QIO                     ;
                             0D            50  E8 00088          BLBS    STATUS, 3$                       ; 0382
                                           50  DD 0008B          PUSHL   STATUS                           ;
                                           7E  D4 0008D          CLRL    -(SP)                            ;
                                000000000G 8F  DD 0008F          PUSHL   #CNF$_SYSID                      ;
                             67            03  FB 00095          CALLS   #3, LIB$SIGNAL                   ;
                             50            01  D0 00098 3$:      MOVL    #1, R0                           ; 0383
                                           04     0009B 4$:      RET                                      ; 0384

; Routine Size:  156 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL  'buffer_id   Remove the system id message from buffer'
1 ROUTINE BUFFER_ID (CIR) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !    Buffer_id is an AST routine called when a read for a system ID on
1 !    the NI completes or cancels.  If the read was cancelled then the
1 !    circuit's buffers are deallocated.  If the read was successful
1 !    then the id message must be removed to another buffer so that the buffer
1 !    can be re-used and the read re-issued immediately.  The circuit block
1 !    and the id message will be queued to the work queue to be parsed and stored.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !        cir     CIRcuit control block.  Contains buffers and information
1 !                on the circuit.
1 !
1 ! IMPLICIT INPUTS:
1 !        NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !        NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !        NONE
1 !
1 ! SIDE EFFECTS:
1 !        NONE
1 !
1 !--
1
2     BEGIN
2     MAP
2         CIR : REF BBLOCK;
2
2     LOCAL
2         SYSIDM : REF BBLOCK;
2
2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
2         $DESCRIPTOR ('buffer_id'));
2
2     !
2     !   Check to see if surveillance has been disabled.
2     !   Since surveillance is disabled by a routine executed from
2     !   the work queue, and this routine is delivered by AST,
2     !   care must be taken to ensure that errors are not introduced
2     !   due to the timing of the AST completion.
2     !
2
2     !
2     !   If the circuit was marked for disable but the AST was delivered
2     !   before the channel was deassigned, then just quit.
2     !
2     IF .CIR [CIR$B_SURVEIL] EQL NMA$C_SUR_DIS
2     THEN
```

```
3           BEGIN
3           EXECUTE (CNF$FREE_VM (%REF(ADRTYP_BUFSIZ), CIR [CIR$L_ADRTYPBUF]));
3           EXECUTE (CNF$FREE_VM (%REF(SYSIDM_BUFSIZ), CIR [CIR$L_SYSIDMBUF]));
3           CIR [CIR$L_ADRTYPBUF] = 0;
3           CIR [CIR$L_SYSIDMBUF] = 0;
3           RETURN TRUE;
3           END;
2
2       !
2       !   The I/O failed, and not because the surveillance was disabled
2       !   and the I/O cancelled, therefore signal the error.
2
2       IF NOT .CIR [CIR$W_IOSB]
2       THEN
3           BEGIN
3           SIGNAL (CNF$_SYSID, 0, .CIR [CIR$W_IOSB]);
3           RETURN TRUE;
3           END;
2
2       !
2       !   Allocate a descriptor and fill it in
2       !
2       EXECUTE (CNF$GET_ZVM (%REF(DSC$C_S_BLN), SYSIDM));
2
2       SYSIDM [DSC$A_POINTER] = .CIR [CIR$L_SYSIDMBUF];
2       SYSIDM [DSC$W_LENGTH] = .CIR [CIR$W_IOSB1];
2
2       !
2       !   Do the work of copying the ID out of the CIR block at the
2       !   the leisure of the work queue.  STORE_ID will have to check
2       !   whether surveillance was disabled on the circuit between now
2       !   and when it finally executes.
2       !
2       WKQ$ADD_WORK_ITEM (STORE_ID, .CIR, .SYSIDM, .CIR [CIR$L_ADRTYPBUF]);
2
2       CIR [CIR$L_SYSIDMBUF] = 0;
2       CIR [CIR$L_ADRTYPBUF] = 0;
2
2       !
2       !   While still at AST level, hurry up and re-issue the READ for the
2       !   System ID messages to reduce the chance of missing any.
2       !
2       CNF$READ_SYSIDM (.CIR);
2
2       RETURN TRUE;
1       END;                            ! Routine buffer_id


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                  45  43  41  52  54  00028 P.AAF:    .ASCII  \TRACE\                    ;
                                      0002D           .BLKB   3
                            00000005  00030 P.AAE:    .LONG   5                          ;
                            00000000' 00034           .ADDRESS P.AAF                     ;
      64  69  5F  72  65  66  66  75  62  00038 P.AAH:    .ASCII  \buffer_id\                ;
                                      00041           .BLKB   3
```

```
                                        00000009  00044 P.AAG:  .LONG   9                                      ;
                                        00000000' 00048         .ADDRESS P.AAH                                 ;



                                                                .PSECT  $CODE$,NOWRT,2

                                            0004 00000 BUFFER_ID:
                                                                .WORD   Save R2                                ; 0386
                          5E            08  C2 00002            SUBL2   #8, SP                                 ;
                                  0000' CF  9F 00005            PUSHAB  P.AAG                                  ; 0426
                                  0000' CF  9F 00009            PUSHAB  P.AAE                                  ; 0425
                                        01  DD 0000D            PUSHL   #1                                     ;
                   0000G  CF            03  FB 0000F            CALLS   #3, CNF$TRACE                          ;
                          52      04    AC  D0 00014            MOVL    CIR, R2                                ; 0440
                          01      0A    A2  91 00018            CMPB    10(R2), #1                             ;
                                        29  12 0001C            BNEQ    1$                                     ;
                                  3C    A2  9F 0001E            PUSHAB  60(R2)                                 ; 0443
                      04  AE            0E  D0 00021            MOVL    #14, 4(SP)                             ;
                                  04    AE  9F 00025            PUSHAB  4(SP)                                  ;
                   0000G  CF            02  FB 00028            CALLS   #2, CNF$FREE_VM                        ;
                          68            50  E9 0002D            BLBC    STATUS, 3$                             ;
                                  38    A2  9F 00030            PUSHAB  56(R2)                                 ; 0444
                      04  AE      40    8F  9A 00033            MOVZBL  #64, 4(SP)                             ;
                                  04    AE  9F 00038            PUSHAB  4(SP)                                  ;
                   0000G  CF            02  FB 0003B            CALLS   #2, CNF$FREE_VM                        ;
                          55            50  E9 00040            BLBC    STATUS, 3$                             ;
                                  38    A2  7C 00043            CLRQ    56(R2)                                 ; 0446
                                            04 00046            RET                                            ; 0447
                          14      0C    A2  E8 00047 1$:        BLBS    12(R2), 2$                             ; 0454
                          7E      0C    A2  32 0004B            CVTWL   12(R2), -(SP)                          ; 0457
                                        7E  D4 0004F            CLRL    -(SP)                                  ;
                             00000000G  8F  DD 00051            PUSHL   #CNF$_SYSID                            ;
               00000000G  00            03  FB 00057            CALLS   #3, LIB$SIGNAL                         ;
                                            04 0005E            RET                                            ; 0458
                                  04    AE  9F 0005F 2$:        PUSHAB  SYSIDM                                 ; 0464
                      04  AE            08  D0 00062            MOVL    #8, 4(SP)                              ;
                                  04    AE  9F 00066            PUSHAB  4(SP)                                  ;
                   0000G  CF            02  FB 00069            CALLS   #2, CNF$GET_ZVM                        ;
                          27            50  E9 0006E            BLBC    STATUS, 3$                             ;
                          50      04    AE  D0 00071            MOVL    SYSIDM, R0                             ; 0466
                      04  A0      38    A2  D0 00075            MOVL    56(R2), 4(R0)                          ;
                          60      0E    A2  B0 0007A            MOVW    14(R2), (R0)                           ; 0467
                                  3C    A2  DD 0007E            PUSHL   60(R2)                                 ; 0475
                                        50  DD 00081            PUSHL   R0                                     ;
                                        52  DD 00083            PUSHL   R2                                     ;
                                  0000V CF  9F 00085            PUSHAB  STORE_ID                               ;
                   0000G  CF            04  FB 00089            CALLS   #4, WRQ$ADD_WORK_ITEM                  ;
                                  38    A2  7C 0008E            CLRQ    56(R2)                                 ; 0477
                                        52  DD 00091            PUSHL   R2                                     ; 0484
                    FECC  CF            01  FB 00093            CALLS   #1, CNF$READ_SYSIDM                    ;
                                            04 00098 3$:        RET                                            ; 0487

; Routine Size:  153 bytes,    Routine Base:  $CODE$ + 009C
```

```
1 %SBTTL  'store_id   Parse and store the system id message'
1 ROUTINE STORE_ID (CIR, SYSIDM_DSC, ADRTYPBUF) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !     Store_id is executed off the work queue.
1 !     The id message must be parsed and stored.
1 !     If it is the id from a familiar device then the cell for
1 !     that device's data is updated, otherwise a new cell is created.
1 !
1 ! FORMAL ?ARAMETERS:
1 !
1 !           cir           CIRcuit control block.  Contains buffers and information
1 !                         on the circuit.
1 !
1 !           sysidm_dsc    Descriptor of buffer containing system ID message
1 !
1 !           adrtypbuf     Bufrer containing the current NI address and
1 !                         protocol type
1 !
1 ! IMPLICIT INPUTS:
1 !           NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !           NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !           NONE
1 !
1 ! SIDE EFFECTS:
1 !           NONE
1 !
1 !--
1
2     BEGIN
2     MAP
2         ADRTYPBUF : REF BBLOCK,
2         CIR : REF BBLOCK,
2         SYSIDM_DSC : REF BBLOCK;
2     LOCAL
2         SID : REF BBLOCK,               ! System Id storage block
2         SIMBUF;
2
2
2     CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE'),
2         $DESCRIPTOR ('store_id'));
2
2     !
2     !   If the circuit was not marked for disable after the AST was delivered
2     !   but before STORE_ID was called from the work queue then store the
2     !   system ID message in the circuit block.
2     !
2     IF .CIR [CIR$B_SURVEIL] EQL NMA$C_SUR_ENA
2     THEN
3         BEGIN
```

```
3
3              !
3              !    Allocate a buffer to hold the system ID message.
3              !
3              EXECUTE (CNF$GET_ZVM (%REF(SID$C_LENGTH), SID) );
3              EXECUTE ($GETTIM ( TIMADR = SID [SID$Q_LSTREPORT] ));
3
3              !
3              !        Extract Current NI address and Protocol type from
3              !        the ADRTYPBUF buffer and deallocate the buffer.
3              !
3              CH$MOVE (SID$C_ADRLEN, .ADRTYPBUF + SID$C_ADRLEN, SID [SID$T_CURADR]);
3
3              !
3              !        If TYPE is not Protocol 260 (Remote Console system ID)
3              !        Then there's been an error.
3              !
3 !!           TBS
3
3              !
3              !        Extract and store info from the System ID Message buffer
3              !
3              SIMBUF = .SYSIDM_DSC [DSC$A_POINTER] + 4;       ! Point to beginning of buffer and skip past
3                                                              !  Code byte, Pad byte and Receipt number word
3              !
3              !   The system ID message is self describing.  A word of type code
3              !   is following by a word containing the length in bytes of the data
3              !   field which follows.  Loop reading a type word, length byte and
3              !   processing the data field, until the whole message has been
3              !   stored away in the appropriate fields of the SID which will be
3              !   placed in the CIR block.
3              !
3              WHILE .SIMBUF LSS (.SYSIDM_DSC [DSC$A_POINTER] + .SYSIDM_DSC [DSC$W_LENGTH]) DO
4                  BEGIN
4                  LOCAL
4                      LENGTH,
4                      TYPE;
4
4                  TYPE = .(.SIMBUF) <0, 16>;          ! Word of type code
4                  LENGTH = .(.SIMBUF) <16, 8>;        ! Byte of length of data field
4                  IF .LENGTH LEQ 0
4                  THEN
5                      BEGIN
5                      CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
5                          $DESCRIPTOR ('store_id - illegal system ID field length'));
5                      EXITLOOP;
4                      END;
4
4                  SIMBUF = .SIMBUF + 3;               ! Skip type and length to data
4
4                  !
4                  !    Dispatch on field type to store data in proper location in
4                  !    SID block.
4                  !
4                  SELECTONE .TYPE OF
4                      SET
4
```

```
4                  !
4                  !    MOP version, MOP version ECO, MOP version User ECO
4                  !
4                  [SIM$C_MOPVERTYP]:
5                      BEGIN
5                      IF .LENGTH EQL 3 THEN
6                          BEGIN
6                          SID [SID$B_MOPVER] = .(.SIMBUF) <0, 8>;         ! MOP version
6                          SID [SID$B_MOPECO] = .(.SIMBUF) <8, 16>;        ! MOP version ECO
6                          SID [SID$B_MOPUSRECO] = .(.SIMBUF) <16, 8>;     ! MOP version User ECO
6                          END
5                      ELSE
5                          CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
5                              $DESCRIPTOR ('store_id - MOP'));
5                      SIMBUF = .SIMBUF + .LENGTH;                 ! Skip over data to next TYPE
4                      END;
4
4                  !
4                  !   Funtions are returned in a word bit mask.  To make
4                  !   life easier when building the SHOW response later,
4                  !   figure how many of the bits in the mask are set.
4                  !
4                  [SIM$C_FUNCTNTYP]:
5                      BEGIN
5                      BIND
5                          FUNCTIONS = SID [SID$W_FUNCTIONS] : BITVECTOR [SID$C_MAXFUNC];
5
5                      IF .LENGTH NEQ 2
5                      THEN
5                          CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
5                              $DESCRIPTOR ('store_id - FUNCTION LENGTH'));
5                      SID [SID$W_FUNCTIONS] = .(.SIMBUF) <0, 16>;
5                      SIMBUF = .SIMBUF + .LENGTH;                 ! Skip over data to next TYPE
5
5                      !
5                      !   Count the number of bits that are set
5                      !
5                      INCR INDEX FROM 0 TO SID$C_MAXFUNC - 1 DO
5                          IF .FUNCTIONS [.INDEX]
5                          THEN SID [SID$B_NUMFUNC] = .SID [SID$B_NUMFUNC] + 1;
4                      END;
4
4                  !
4                  !   Get the 6-byte Hardware address
4                  !
4                  [SIM$C_HDWADRTYP]:
5                      BEGIN
5                      CH$MOVE (SID$C_ADRLEN, .SIMBUF, SID [SID$T_HRDWADR]);
5                      SIMBUF = .SIMBUF + .LENGTH;                 ! Skip over data to next TYPE
4                      END;
4
4                  !
4                  !   Get the device type
4                  !
4                  [SIM$C_DEVICETYP]:
5                      BEGIN
5                      SID [SID$B_DEVICE] = .(.SIMBUF) <0, 8>;
```

```
                                SIMBUF = .SIMCUF + .LENGTH;                     ! Skip over data to next TYPE
                                END;

                            !
                            !   There is a data field returned which we don't understand.
                            !   Skip over it and proceed merrily on our way.
                            !
                            [OTHERWISE]:
                                BEGIN
                                CNF$TRACE (DBG$C_TRACE, $DESCRIPTOR('TRACE *** ERROR '),
                                    $DESCRIPTOR ('store_id'));
                                SIMBUF = .SIMBUF + .LENGTH;                     ! Skip over data to next TYPE
                                END;

                            TES;

                        END;                                    ! While parsing System ID Message buffer

                    !
                    !   Place the SID into the circuit block on an ordered linked list
                    !
                    INSERT_SID (.SID, .CIR);
                    END;                                        ! If surveillance was still enabled

                !
                !   If the circuit was marked for disable after the AST was delivered
                !   but before STORE_ID was called from the work queue then just
                !   deallocate the buffers and quit.
                !
                EXECUTE (CNF$FREE_VM (%REF(ADRTYP_BUFSIZ), ADRTYPBUF) );
                EXECUTE (CNF$FREE_VM (%REF(SYSIDM_BUFSIZ), SYSIDM_DSC [DSC$A_POINTER]) );
                EXECUTE (CNF$FREE_VM (%REF(DSC$C_S_BLN), SYSIDM_DSC));

                RETURN TRUE;
                END;                            ! Routine Store_id


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                  45  43  41  52  54  0004C P.AAJ:  .ASCII  \TRACE\                                     ;
                                                      00051         .BLKB   3
                                            00000005  00054 P.AAI:  .LONG   5                                           ;
                                            00000000' 00058         .ADDRESS P.AAJ                                      ;
                      64  69  5F  65  72  6F  74  73  0005C P.AAL:  .ASCII  \store_id\                                  ;
                                            00000008  00064 P.AAK:  .LONG   8                                           ;
                                            00000000' 00068         .ADDRESS P.AAL                                      ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  0006C P.AAN:  .ASCII  \TRACE *** ERROR \                    ;
                                                      20  0007B                                                         ;
                                            00000010  0007C P.AAM:  .LONG   16                                          ;
                                            00000000' 00080         .ADDRESS P.AAN                                      ;
65  6C  6C  69  20  2D  20  64  69  5F  65  72  6F  74  73  00084 P.AAP:  .ASCII  \store_id - illegal system ID field lengt\  ;
66  20  44  49  20  6D  65  74  73  79  73  20  6C  61  67  00093                                                         ;
                74  67  6E  65  6C  20  64  6C  65  69  000A2                                                         ;
                                                      68  000AC         .ASCII  \h\                                     ;
                                                          000AD         .BLKB   3
                                            00000029  000B0 P.AAO:  .LONG   41                                          ;
                                            00000000'  000B4           .ADDRESS P.AAP                              ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  000B8 P.AAR:   .ASCII  \TRACE *** ERROR \                  ;
                                                       20  000C7                                                   ;
                                            00000010   000C8 P.AAQ:    .LONG   16                                  ;
                                            00000000'  000CC           .ADDRESS P.AAR                              ;
    50  4F  4D  20  2D  20  64  69  5F  65  72  6F  74  73  000D0 P.AAT:   .ASCII  \store_id - MOP\                    ;
                                                       000DE           .BLKB   2                                   ;
                                            0000000E   000E0 P.AAS:    .LONG   14                                  ;
                                            00000000'  000E4           .ADDRESS P.AAT                              ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  000E8 P.AAV:   .ASCII  \TRACE *** ERROR \                  ;
                                                       20  000F7                                                   ;
                                            00000010   000F8 P.AAU:    .LONG   16                                  ;
                                            00000000'  000FC           .ADDRESS P.AAV                              ;
43  4E  55  46  20  2D  20  64  69  5F  65  72  6F  74  73  00100 P.AAX:   .ASCII  \store_id - FUNCTION LENGTH\        ;
                48  54  47  4E  45  4C  20  4E  4F  49  54  0010F                                                   ;
                                                       0011A           .BLKB   2                                   ;
                                            0000001A   0011C P.AAW:    .LONG   26                                  ;
                                            00000000'  00120           .ADDRESS P.AAX                              ;
52  4F  52  52  45  20  2A  2A  2A  20  45  43  41  52  54  00124 P.AAZ:   .ASCII  \TRACE *** ERROR \                  ;
                                                       20  00133                                                   ;
                                            00000010   00134 P.AAY:    .LONG   16                                  ;
                                            00000000'  00138           .ADDRESS P.AAZ                              ;
                            64  69  5F  65  72  6F  74  73  0013C P.ABB:   .ASCII  \store_id\                          ;
                                            00000008   00144 P.ABA:    .LONG   8                                   ;
                                            00000000'  00148           .ADDRESS P.ABB                              ;

                                                                       .EXTRN  SYS$GETTIM

                                                                       .PSECT  $CODE$,NOWRT,2

                                                 0FFC  00000 STORE_ID:
                                                                       .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11   ; 0489
                               5B       0000'   CF   9E  00002           MOVAB   P.AAK, R11                          ;
                               5E                08   C2  00007           SUBL2   #8, SP                              ;
                                                 5B   DD  0000A           PUSHL   R11                                 ; 0535
                                        F0       AB   9F  0000C           PUSHAB  P.AAI                               ; 0534
                                                 01   DD  0000F           PUSHL   #1                                  ;
                         0000G         CF        03   FB  00011           CALLS   #3, CNF$TRACE                       ;
                               5A       04       AC   D0  00016           MOVL    CIR, R10                            ; 0542
                                        0A       AA   95  0001A           TSTB    10(R10)                             ;
                                                 03   13  0001D           BEQL    1$                                  ;
                                               00E3   31  0001F           BRW     19$                                 ;
                                        04       AE   9F  00022 1$:       PUSHAB  SID                                 ; 0549
                          04   AE                25   D0  00025           MOVL    #37, 4(SP)                          ;
                                        04       AE   9F  00029           PUSHAB  4(SP)                               ;
                         0000G         CF        02   FB  0002C           CALLS   #2, CNF$GET_ZVM                     ;
                               0E                50   E9  00031           BLBC    STATUS, 2$                          ;
                               56       04       AE   D0  00034           MOVL    SID, R6                             ; 0550
                                        16       A6   9F  00038           PUSHAB  22(R6)                              ;
                     00000000G         00        01   FB  0003B           CALLS   #1, SYS$GETTIM                      ;
                               01                50   E8  00042 2$:       BLBS    STATUS, 3$                          ;
                                                      04  00045           RET                                         ;
                               50       0C       AC   D0  00046 3$:       MOVL    ADRTYPBUF, R0                       ; 0556
            10      A6         06       A0       06   28  0004A           MOVC3   #6, 6(R0), 16(R6)                   ;
                               59       08       AC   D0  00050           MOVL    SYSIDM_DSC, R9                      ; 0567
                  58           04       A9       04   C1  00054           ADDL3   #4, 4(R9), SIMBUF                   ;
                               50                69   3C  00059 4$:       MOVZWL  (R9), R0                            ; 0577
                         50      04   A9  C0 0005C           ADDL2   4(R9), R0                          ;
                         50           58  D1 00060           CMPL    SIMBUF, R0                         ;
                                      16  18 00063           BGEQ    5$                                 ;
                         52           68  3C 00065           MOVZWL  (SIMBUF), TYPE                     ; 0583
                         57      02   A8  9A 00068           MOVZBL  2(SIMBUF), LENGTH                  ; 0584
                                      0F  14 0006C           BGTR    6$                                 ; 0585
                                 4C   AB  9F 0006E           PUSHAB  P.AAO                              ; 0589
                                 18   AB  9F 00071           PUSHAB  P.AAM                              ; 0588
                                      01  DD 00074           PUSHL   #1                                 ;
                  0000G  CF           03  FB 00076           CALLS   #3, CNF$TRACE                      ;
                                      7F  11 0007B 5$:       BRB     18$                                ; 0587
                         58           03  C0 0007D 6$:       ADDL2   #3, SIMBUF                         ; 0593
                         01           52  D1 00080           CMPL    TYPE, #1                           ; 0605
                                      18  12 00083           BNEQ    8$                                 ;
                         03           57  D1 00085           CMPL    LENGTH. #3                         ; 0607
                                      0B  12 00088           BNEQ    7$                                 ;
                  1E     A6           68  B0 0008A           MOVW    (SIMBUF), 30(R6)                   ; 0609
                  20     A6      02   A8  90 0008E           MOVB    2(SIMBUF), 32(R6)                  ; 0611
                                      62  11 00093           BRB     17$                                ; 0607
                                 7C   AB  9F 00095 7$:       PUSHAB  P.AAS                              ; 0615
                                 64   AB  9F 00098           PUSHAB  P.AAQ                              ; 0614
                                      53  11 0009B           BRB     16$                                ;
                         02           52  D1 0009D 8$:       CMPL    TYPE, #2                           ; 0624
                                      2B  12 000A0           BNEQ    13$                                ;
                         02           57  D1 000A2           CMPL    LENGTH, #2                         , 0629
                                      0F  13 000A5           BEQL    9$                                 ;
                               0088   CB  9F 000A7           PUSHAB  P.AAW                              ; 0632
                               0094   CB  9F 000AB           PUSHAB  P.AAU                              ; 0631
                                      01  DD 000AF           PUSHL   #1                                 ;
                  0000G  CF           03  FB 000B1           CALLS   #3, CNF$TRACE                      ;
                  22     A6           68  B0 000B6 9$:       MOVW    (SIMBUF), 34(R6)                   ; 0633
                         58           57  C0 000BA           ADDL2   LENGTH, SIMBUF                     ; 0634
                                      50  D4 000BD           CLRL    INDEX                              ; 0639
          03      22     A6           50  E1 000BF 10$:      BBC     INDEX, 34(R6), 11$                 ; 0640
                                 21   A6  96 000C4           INCB    33(R6)                             ; 0641
          F4             50           0F  F3 000C7 11$:      AOBLEQ  #15, INDEX, 10$                    ; 0640
                                      8C  11 000CB 12$:      BRB     4$                                 ; 0599
                         07           52  D1 000CD 13$:      CMPL    TYPE, #7                           ; 0647
                                      07  12 000D0           BNEQ    14$                                ;
     0A   A6             68           06  28 000D2           MOVC3   #6, (SIMBUF), 10(R6)               ; 0649
                                      1E  11 000D7           BRB     17$                                ; 0650
               00000064  8F           52  D1 000D9 14$:      CMPL    TYPE, #100                         ; 0656
                                      06  12 000E0           BNEQ    15$                                ;
                  24     A6           68  90 000E2           MOVB    (SIMBUF), 36(R6)                   ; 0658
                                      0F  11 000E6           BRB     17$                                ; 0659
                               00E0   CB  9F 000E8 15$:      PUSHAB  P.ABA                              ; 0669
                               00D0   CB  9F 000EC           PUSHAB  P.AAY                              ; 0668
                                      01  DD 000F0 16$:      PUSHL   #1                                 ;
                  0000G  CF           03  FB 000F2           CALLS   #3, CNF$TRACE                      ;
                         58           57  C0 000F7 17$:      ADDL2   LENGTH, SIMBUF                     ; 0670
                                      CF  11 000FA           BRB     12$                                ; 0577
                               0440   8F  BB 000FC 18$:      PUSHR   #^M<R6,R10>                        ; 0680
                  0000V  CF           02  FB 00100           CALLS   #2, INSERT_SID                     ;
                                 0C   AC  9F 00105 19$:      PUSHAB  ADRTYPBUF                          ; 0688
                  04     AE           0E  D0 00108           MOVL    #14, 4(SP)                         ;
                                 04   AE  9F 0010C           PUSHAB  4(SP)                              ;
                  0000G  CF           02  FB 0010F           CALLS   #2, CNF$FREE_VM                    ;
```

```
                           24              50 E9 00114          BLBC    STATUS, 20$                           ;
              7E     08    AC              04 C1 00117          ADDL3   #4, SYSIDM_DSC, -(SP)                 ; 0689
                     04    AE        40    8F 9A 0011C          MOVZBL  #64, 4(SP)                            ;
                                     04    AE 9F 00121          PUSHAB  4(SP)                                 ;
                  0000G    CF              02 FB 00124          CALLS   #2, CNF$FREE_VM                       ;
                           0F              50 E9 00129          BLBC    STATUS, 20$                           ;
                                     08    AC 9F 0012C          PUSHAB  SYSIDM_DSC                            ; 0690
                     04    AE              08 D0 0012F          MOVL    #8, 4(SP)                             ;
                                     04    AE 9F 00133          PUSHAB  4(SP)                                 ;
                  0000G    CF              02 FB 00136          CALLS   #2, CNF$FREE_VM                       ;
                                              04 0013B 20$:     RET                                           ; 0693

; Routine Size:  316 bytes,    Routine Base:  $CODE$ + 0135
```

```
1 %SBTTL  'insert_sid  Insert SID into CIR block'
1 ROUTINE INSERT_SID (SID, CIR) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !  Place the system ID message into a doubly linked list of system ID
1 !  messages in the CIR block, ordered by NI hardware address.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       sid     System ID storage block.
1 !
1 !       cir     CIRcuit control block.  Contains buffers and information
1 !               on the circuit.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       Success or a failure returned by a routine which is called
1 !       from INSERT_SID
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2    BEGIN
2    MAP
2        CIR : REF BBLOCK,
2        SID : REF BBLOCK;
2
2    LOCAL
2        STATUS;
2
2
2    IF .CIR [CIR$L_SIDFLINK] EQL CIR [CIR$L_SIDFLINK]
2    THEN
2        !
2        !    This is the first ID message stored, so just plop it in.
2        !
2        INSQUE (.SID, .CIR [CIR$L_SIDBLINK])
2    ELSE
3        BEGIN
3        !
3        !    Create a doubly linked list ordered by NI Hardware address
3        !
3        LOCAL
3            CMPR_ADR_DSC : BBLOCK [DSC$C_S_BLN],
3            NSRT_ADR_DSC : BBLOCK [DSC$C_S_BLN],
3            CMP_SID : REF BBLOCK;
3
```

```
3          !
3          !    Build a descriptor of the NI hardware address to be used in
3          !    a string compare utility routine.
3          !
3          CH$FILL (0, DSC$C_S_BLN, NSRT_ADR_DSC);
3          NSRT_ADR_DSC [DSC$W_LENGTH] = SID$C_ADRLEN;
3          NSRT_ADR_DSC [DSC$A_POINTER] = SID [SID$T_HRDWADR];
3          CH$FILL (0, DSC$C_S_BLN, CMPR_ADR_DSC);
3          CMPR_ADR_DSC [DSC$W_LENGTH] = SID$C_ADRLEN;
3
3          !
3          !    Run thru the whole list
3          !
3          CMP_SID = .CIR [CIR$L_SIDFLINK];          ! SID being compared against
3          WHILE .CMP_SID NEQ CIR [CIR$L_SIDFLINK] DO
4              BEGIN
4              CMPR_ADR_DSC [DSC$A_POINTER] = CMP_SID [SID$T_HRDWADR];
4
4              CASE STR$COMPARE (NSRT_ADR_DSC, CMPR_ADR_DSC)
4              FROM -1 TO 1 OF      ! Either Less than, Equal to, or Greater than
4                  SET
4
4                  [-1]:
4                      !
4                      !    Less than: Insert here
4                      !
5                      BEGIN
5                      INSQUE (.SID, .CMP_SID [SID$L_BLINK]);
5                      RETURN TRUE;
4                      END;
4
4                  [0] :
4                      !
4                      !    Equal to: Replace with the new message
4                      !
5                      BEGIN
5                      REMQUE (.CMP_SID, STATUS);
5                      INSQUE (.SID, .CMP_SID [SID$L_BLINK]);
5                      EXECUTE (CNF$FREE_VM (%REF(SID$C_LENGTH), CMP_SID) );
5                      RETURN TRUE;
4                      END;
4
4                  [1] :
4                      !
4                      !    It's greater than: Keep moving down the list
4                      !
5                      BEGIN
5                      CMP_SID = .CMP_SID [SID$L_LINK];
4                      END;
4
4                  TES;
3              END;                     ! End WHILE traversing list of SYSTEM IDs
3
3          !
3          !    It was greater than all the ones in the list so we dropped
3          !    down to here where it should be inserted at end of the list
3          !
```

```
;  627          0808  3          INSQUE (.SID, .CIR [CIR$L_SIDBLINK])
;  628          0809  3
;  629          0810  2          END;                        ! ELSE the list was not empty
;  630          0811  2
;  631          0812  2       RETURN TRUE;
;  632          0813  1       END;                           ! Routine insert_sid



                                                    007C 00000 INSERT_SID:
                                                                        .WORD    Save R2,R3,R4,R5,R6                          ; 0695
                          5E              18  C2 00002                  SUBL2    #24, SP
                          56        08    AC  D0 00005                  MOVL     CIR, R6                                      ; 0735
                          50        40    A6  9E 00009                  MOVAB    64(R6), R0                                   ;
                          50        40    A6  D1 0000D                  CMPL     64(R6), R0                                   ;
                                          07  12 00011                  BNEQ     1$                                           ;
                  44      B6        04    BC  0E 00013                  INSQUE   @SID, @68(R6)                                ; 0740
                                          58  11 00018                  BRB      5$                                           ;
    08        00          6E              00  2C 0001A  1$:             MOVC5    #0, (SP), #0, #8, NSRT_ADR_DSC               ; 0755
                                    08    AE     0001F                                                                        ;
                  08      AE              06  B0 00021                  MOVW     #6, NSRT_ADR_DSC                             ; 0756
           0C     AE      04      AC      0A  C1 00025                  ADDL3    #10, SID, NSRT_ADR_DSC+4                     ; 0757
    08            00      6E              00  2C 0002B                  MOVC5    #0, (SP), #0, #8, CMPR_ADR_DSC               ; 0758
                                    10    AE     00030                                                                        ;
                  10      AE              06  B0 00032                  MOVW     #6, CMPR_ADR_DSC                             ; 0759
                  04      AE        40    A6  D0 00036                  MOVL     64(R6), CMP_SID                              ; 0764
                          52        04    AE  D0 0003B  2$:             MOVL     CMP_SID, R2                                  ; 0765
          50      08      AC  00000040    8F  C1 0003F                  ADDL3    #64, CIR, R0                                 ;
                          50              52  D1 00048                  CMPL     R2, R0                                       ;
                                          4C  13 0004B                  BEQL     8$                                           ;
                  14      AE        0A    A2  9E 0004D                  MOVAB    10(R2), CMPR_ADR_DSC+4                       ; 0767
                                    10    AE  9F 00052                  PUSHAB   CMPR_ADR_DSC                                 ; 0769
                                    0C    AE  9F 00055                  PUSHAB   NSRT_ADR_DSC                                 ;
            00000000G     00              02  FB 00058                  CALLS    #2, STR$COMPARE                              ;
         02 FFFFFFFF      8F              50  CF 0005F                  CASEL    R0, #-1, #2                                  ;
        002C          000D          0006         00067  3$:             .WORD    4$-3$,-                                      ;
                                                                                 6$-3$,-                                      ;
                                                                                 7$-3$                                        ;
                  04      B2        04    BC  0E 0006D  4$:             INSQUE   @SID, @4(R2)                                 ; 0778
                                          2E  11 00072  5$:             BRB      9$                                           ; 0779
                          53              62  0F 00074  6$:             REMQUE   (R2), STATUS                                 ; 0787
                          50        04    AE  D0 00077                  MOVL     CMP_SID, R0                                  ; 0788
                  04      B0        04    BC  0E 0007B                  INSQUE   @SID, @4(R0)                                 ;
                                    04    AE  9F 00080                  PUSHAB   CMP_SID                                      ; 0789
                  04      AE              25  D0 00083                  MOVL     #37, 4(SP)                                   ;
                                    04    AE  9F 00087                  PUSHAB   4(SP)                                        ;
               0000G      CF              02  FB 0008A                  CALLS    #2, CNF$FREE_VM                              ;
                          10              50  E8 0008F                  BLBS     STATUS, 9$                                   ;
                                              04 00092                  RET                                                   ; 0790
                  04      AE              62  D0 00093  7$:             MOVL     (R2), CMP_SID                                ; 0798
                                          A2  11 00097                  BRB      2$                                           ; 0765
                          50        08    AC  D0 00099  8$:             MOVL     CIR, R0                                      ; 0808
                  44      B0        04    BC  0E 0009D                  INSQUE   @SID, @68(R0)                                ;
                          50              01  D0 000A2  9$:             MOVL     #1, R0                                       ; 0812
                                              04 000A5                  RET                                                   ; 0813
```

```
; Routine Size:  166 bytes,    Routine Base:  $CODE$ + 0271
;  634          0814  1 END                              ! End of module CNFSTORE
;  635          0815  0 ELUDOM



                                                          .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                  Bytes                       Attributes
;
;  $PLIT$                         332  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         791  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                        0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)



;                          Library Statistics
;
;                                         -------- Symbols --------      Pages       Processing
;       File                              Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1       9776         9         0        581        00:01.0
;  _$255$DUA28:[SHRLIB]NMALIBRY.L32;1       887         2         0         47        00:00.8



;                               COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CNFSTORE/OBJ=OBJ$:CNFSTORE MSRC$:CNFSTORE/UPDATE=(ENH$:CNFSTORE)

; Size:         791 code + 332 data bytes
; Run Time:        00:18.2
; Elapsed Time:    00:36.4
; Lines/CPU Min:    2692
; Lexemes/CPU-Min: 19912
; Memory Used:  155 pages
; Compilation Complete
```

0280 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CNFSEND
LIS

NPADEF
MDL

CNFSTORE
LIS

NMLSHR
MAP

NMAHEAD
B32

NPAMAC
MAR

CNFSHOW
LIS

CNFWQDEF
LIS

NMATAIL
B32

NML

NMLDDL
B32

NML
MAP

NMLDEF
MDL

NMAFILES
LIS

NMAFIELDS
LIS

CNFWORKQ
LIS